International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Need for Intelligent Software Cost Estimation and its Methods: A Comprehensive Overview

M. S. Rekha[1*], Bharathi Malakreddy A.[2]

Submitted: 08/01/2024 **Revised**: 14/02/2024 **Accepted**: 22/02/2024

Abstract: Software cost estimation stands as a critical phase in the software development life cycle, impacting budgeting, resource allocation, and project planning. The dynamic and complex nature of software projects, along with the rapid evolution of technology, necessitates an intelligent approach to cost estimation. This paper presents a comprehensive overview of the need for intelligent software cost estimation and delves into various methods employed to achieve accuracy and efficiency in estimations. It examines traditional models, such as COCOMO and Function Point Analysis, alongside modern techniques that leverage machine learning and artificial intelligence to adapt to the complexities of software projects. By comparing the effectiveness, challenges, and applicability of each method, the paper highlights the evolution of cost estimation practices and the growing importance of incorporating intelligence into these processes. It concludes with insights into future directions, emphasizing the integration of predictive analytics and data-driven decision-making in improving the reliability and precision of software cost estimations. Through this overview, the paper aims to provide stakeholders with a deeper understanding of intelligent cost estimation methods, facilitating better planning and management of software projects.

***Keywords**: Cost estimation; Project management; FPA; COCOMO*

1. Introduction

Furthermore, our research endeavour tackles the various challenges associated with various circumstances, such as the intricacy of the estimation problem, the challenge of gathering data, and the absence of reliable cost models. It identifies areas that need further research and makes suggestions for improving the accuracy of software cost prediction. Apart from providing a pragmatic perspective, the work enumerates case studies and illustrations that demonstrate the practical use of various cost estimation methodologies. With the assistance of this review, our work provides an assessment of the current status of software cost estimation and makes recommendations for future advancements. A set of best practises, guidelines, and methods that have been shown to be successful in predicting software development costs should serve as the foundation for this methodology. [3] A thorough understanding of project management, software engineering, statistical analysis, and software development processes is necessary for the creation of a software cost estimation technique. The technique should consider the complexity and breadth of the software project and be flexible enough to work in a variety of software development contexts, including waterfall, agile, and hybrid. [4] In addition, this essay reviews the study's subtopics, which are listed below.

1.1 Software cost estimation model

The cost of producing a software application is estimated using mathematical models called software cost estimating models. [5] Project managers can utilise these models to make well-informed decisions regarding the resources, time, and cost required to finish a software development project. The Function Point Analysis (FPA), COCOMO, and Delphi Method are now the most widely utilised cost estimation methodologies. (6) An estimation approach called COCOMO (Constructive Cost approach) makes use of a number of parameters to forecast the price of a software project. The size, complexity, and kind of the software determine the parameters.

*[1]Research Scholar, Department of Computer Science and Engineering,
B M S Institute of Technology & Management Yelahanka affiliated to Visvesvaraya Technological University, Belagavi-590018, Karnataka, India
*Assistant Professor, Department of CSE, R L Jalappa Institute of Technology,
Doddaballapur-561203, India rekha061264@gmail.com
[2]Professor, Department of Artificial Intelligence and Machine Learning, B M S Institute of Technology & Management, Yelahanka, Karnataka, India., bharathi_m@bmsit.in*

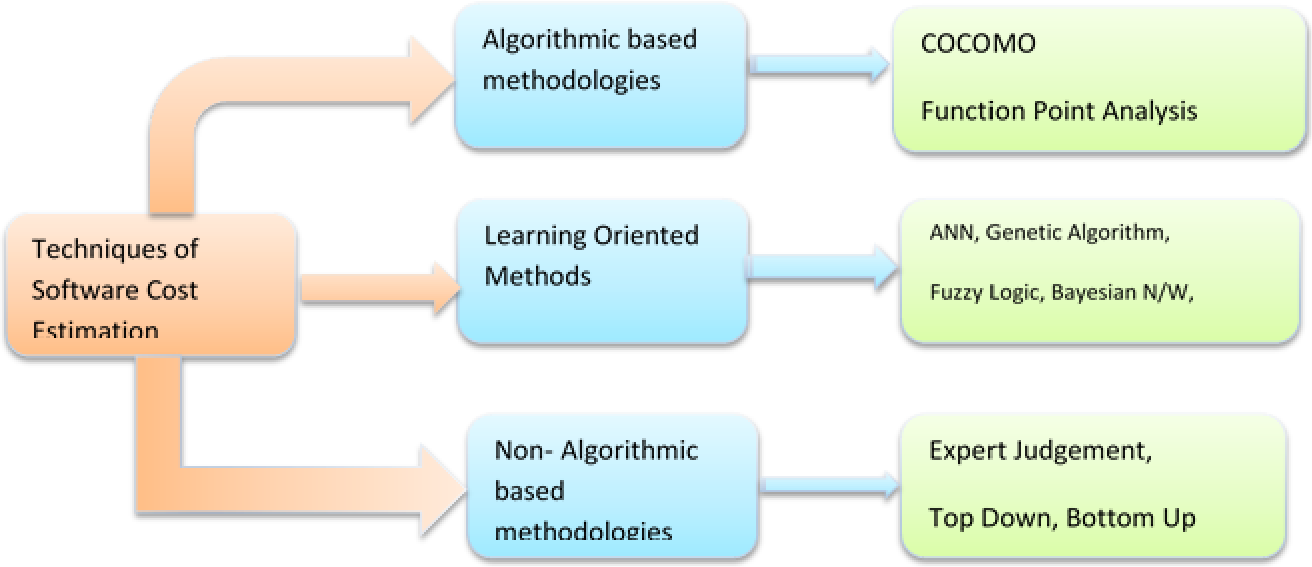


Fig 1: Software cost estimation model

1.2 Model structure and components

Input and output components are the two general categories into which model structure components are separated. The data that is utilised to construct the model is provided by the input components, while the model's output is produced by the output components. A model structure's constituent parts can be connected to one another in a number of methods, including relationships, data flows, and equations. [10–8]

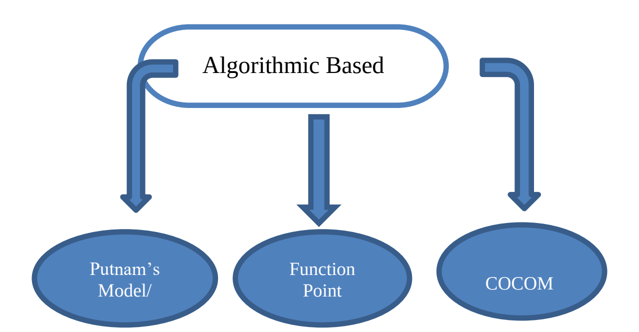
A model structure's constituent parts can also be arranged in many ways. They can be divided into groups according to categories like intermediate, output, and input components. {11] Subcategories, such as variables, equations, and functions, can then be created from these main categories. It is also possible to arrange the elements of a model structure in a hierarchical structure, in which the elements at lower levels are more specialised and those at higher levels are more general.To sum up, a model structure is a kind of abstract representation of a system made up of several components that stand in for different system elements. The components in question may comprise of input, output, and intermediate elements. A model structure's constituent parts can be arranged in a number of ways, including categories, subcategories, and hierarchical structures.

1.3 Input variables and metrics

Metrics and input variables are used in software project cost estimation. The input variables include the project's size, complexity, team composition, and timeliness. Common metrics include test cases, user stories, and lines of code. Additional measures include internet services, database items, and user interface displays. These metrics put a number on labour, materials, and travel costs associated with the project. A project cost estimate and a breakdown of the cost components are generated via cost estimating.

The detailed analysis of the three different models in view with the input variables and different parametric basis can be described as follows:

The algorithmic based methods are as shown in figure 2 and it can be utilized with mathematical and historical data in order to predict the costs. Examples include **COCOMO** (Constructive Cost Model), which estimates costs based on project size and complexity parameters. **Function Point Analysis** measures the functionality delivered to users, aiding in cost estimation. **PUTNAM 'S Model/ SLIM** considers various parameters such as lines of code, team productivity, and project requirements to generate reliable estimates for project planning and decision-making in which above parameters will help in refine and optimizing the cost practices.



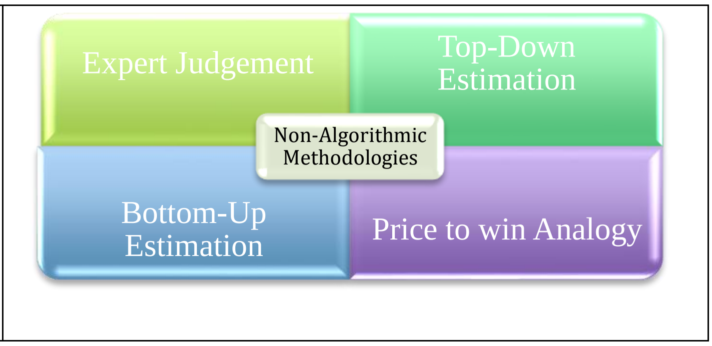


Fig 2: Block Diagram of Algorithmic based

Fig 3: Block Diagram of Non-Algorithmic Methodologies

2. Methodologies

Non-algorithmic-based cost estimation methods are represented in figure 3 which rely on expert judgment, analogies, or historical data without explicitly using mathematical algorithms for estimation. Example include **Expert Judgment** relies on the insights and experience of individuals or a panel of experts to estimate project costs based on their domain knowledge, past experiences, and intuition. **The Top-Down Approach** involves estimating project costs at a higher level of abstraction, typically based on overall project characteristics or historical data. **The Bottom-Up Approach** breaks down project tasks or components into smaller, detailed elements, with costs estimated individually and then aggregated to derive the total project cost.**The Price-to-Win Analogy** refers to estimating project costs based on the competitor's pricing strategies or market benchmarks to develop a competitive bid or proposal, often used in competitive bidding environments such as government contracts or large-scale projects.

The learning-oriented methods are as shown in figure 4, leverage on feedback mechanisms, historical data and refine the iterative development mechanisms for accuracy and reliability of the software cost with respect to time. Examples like **Artificial Neural Networks (ANNs)** are the computational models inspired by the structure and function of biological neural networks. **Genetic Algorithms (GAs)** are optimization algorithms inspired by the process of natural selection and genetics. **Fuzzy Logic** is a mathematical approach to dealing with uncertainty and imprecision in decision-making **Bayesian Networks** are probabilistic graphical models representing uncertain relationships between variables using directed acyclic graphs. **Regression Trees** are non-parametric supervised learning models used for regression tasks. **Support Vector Machine** is a supervised learning algorithm used for classification and regression tasks. SVM aims to find the hyperplane that best separates the classes in the input space by maximizing the margin between the nearest data points

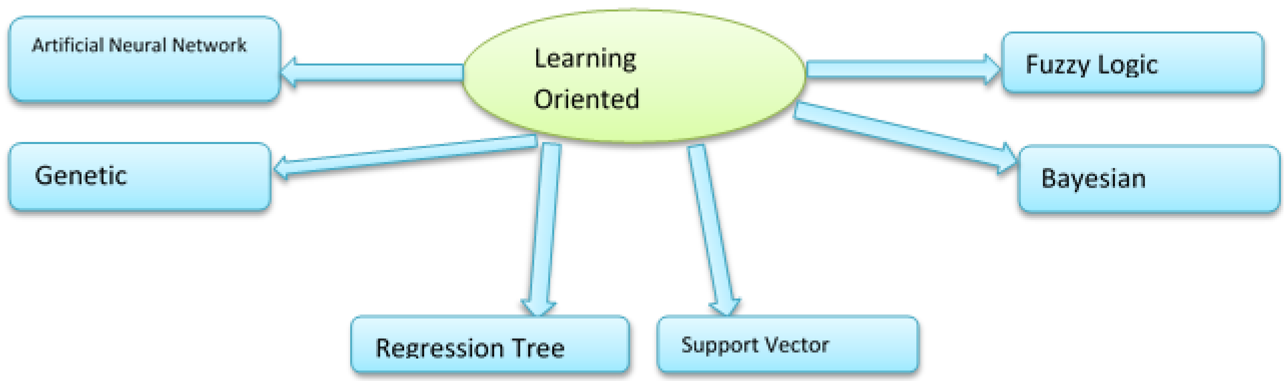


Fig 4: Block diagram of learning-oriented methods

1.4 Problem statement

Any software system's successful development depends on the process of software cost estimation. It entails the steps of determining the resources required to construct the system, estimating the cost of development, and analysing the software system's needs. Because it necessitates a thorough examination of the needs and resources needed to construct the software system, the process of estimating software costs can be extremely difficult. Understanding the development process and related expenses in great depth is also necessary for the cost estimation procedure. The goal of software cost assessment is to precisely ascertain the software system's cost prior to the start of its development, guaranteeing the software system's successful and economical development.

1.5 Objectives

1. Create accurate estimates of software development costs by incorporating the resources, time, and materials necessary to complete the project.
2. Identify and analyze cost drivers to accurately assess the financial impact of software development life cycle activities.
3. Utilize quantitative methods such as parametric modeling, functional point analysis, and cost estimation models to improve the accuracy of software cost estimation.
4. Develop and implement strategies to reduce software development costs without sacrificing quality or project timeline.

3. Literature Review

We took into consideration a number of research studies on the creation and design of software cost estimation, some of which are recent and are listed below.

An overview of current software cost estimation methods is given in this work. It describes several methods for estimating software costs, such as expert judgement, algorithmic, and parametric methods. It also goes over the benefits and drawbacks of each method as well as the difficulties in estimating software costs. Along with reviewing previous studies in the area, the report offers recommendations for additional research. [12] An extensive analysis of software cost estimation methods is presented in this study. It describes a number of current methods, such as expert judgement, algorithmic, and parametric methods. It also covers the drawbacks and benefits of each method as well as the difficulties in estimating software costs. The study also examines previous studies in this area and makes recommendations for additional research. [13]

An overview of the state of software cost estimation approaches is given in this work. It describes several methods for estimating software costs, such as expert judgement, algorithmic, and parametric methods. It also goes over the benefits and drawbacks of each method as well as the difficulties in estimating software costs. Along with reviewing previous studies in the area, the report offers recommendations for additional research. [14] This paper reviews the state of the software cost estimates field as well as the methods that are currently in use. It also covers how software cost estimation methods will develop in the future, taking into account newly developed fields like artificial intelligence and machine learning. Techniques including algorithmic, parametric, machine learning, and expert judgement techniques were included in the review. The review also covered the need for more research in this area as well as a number of issues related to software cost assessment. [15]

A comparative comparison of software cost estimation methods is presented in this research. It examines current methods, such as expert judgement, algorithmic, and parametric methods. It also goes over the benefits and drawbacks of each method as well as the difficulties in estimating software costs. Along with reviewing previous studies in the area, the report offers recommendations for additional research. [16] A thorough analysis of software cost estimation methods is presented in this study. It describes a number of current methods, such as expert judgement, algorithmic, and parametric methods. It also goes over the benefits and drawbacks of each method as well as the difficulties in estimating software costs. Along with reviewing previous studies in the area, the report offers recommendations for additional research. [17]

An extensive review of software cost estimation methods for 2021 and after is given in this study. There includes discussion of a number of cost assessment methods, including expert judgement, algorithmic, parametric, and analogy. The topic of complexity and uncertainty, among other parameters utilised in the estimating process, is also covered in the discussion. Along with discussing the many difficulties that arise when estimating software costs, the authors also suggest a framework for upcoming cost estimation techniques. [18] An extensive overview of software cost estimation methods is given in this work. Numerous cost estimation approaches are covered, including expert judgement, parametric, algorithmic, and analogy models. The topic of complexity and uncertainty, among other parameters utilised in the estimating process, is also covered in the discussion. Along with discussing the many difficulties that arise when estimating software costs, the authors also suggest a framework for upcoming cost estimation techniques. [19]

An extensive review of Agile software cost estimation methods is given in this work. It looks at the many methods employed in Agile development and evaluates the benefits and drawbacks of each. It also describes how Agile software cost estimating may be made more accurate and efficient by utilising these strategies. [20] An extensive analysis of software cost estimation methods utilised in Agile development is presented in this study. It examines the several methods and strategies applied in the estimating process and gives a synopsis of their advantages and disadvantages. It also looks at the effects of the different methods and strategies applied in Agile development. [21]

An overview of cost estimation in Agile software development is given in this document. It examines the several methods and strategies applied in the estimating process and gives a summary of their advantages and disadvantages. It also looks at the effects of the different methods and strategies applied in Agile development. [22]

The literature surveys on methodologies used in finding the software cost estimation a can be seen below

Table 1 survey of different software cost estimation methodologies

Year	Author	Methodology	Pros	Cons
2023	K.D.D.Willis et al. [28]	Bottoms-up estimation	-Provides detailed estimates of costs -Can be used to estimate costs for individual components of the project -Can take into account changes in the project scope or timeline	Can be time consuming to complete -Requires detailed knowledge of the project and its components
2023	R.Glennie et al. [62] & R. Silhavy et.al. [26]	Hidden Markov Models (HMM)	Can model time-series data with hidden states and observations Can capture complex dependencies and patterns in sequential data Can be used for prediction, classification, and segmentation tasks	Requires a significant amount of data and expertise to estimate the model parameters Can be computationally expensive to evaluate, particularly for long sequences Sensitive to the choice of model structure and assumptions about the underlying distribution
2022 2022	E.M.D.S. Favero et.al. [27] R. Khoshfetrat et.al[29]	Analogues estimation COCOMO	-Provides a quantitative approach for estimating project cost, effort and time. -Offers a systematic way of evaluating the project characteristics such as size, complexity and risk. -Allows for early-stage estimation which enables early planning, budgeting and decision-making.	-The model assumes that all projects are similar, which might not be true. -The accuracy of the model is dependent on the input data, which may be difficult to gather. -The model is not well suited for agile development methodologies.
2022	U.K Nath et.al[30]	Agile estimation	-Agile estimating and planning helps teams adapt to changing requirements and priorities. -It allows for continuous feedback and improvement, leading to better project outcomes. -It is flexible, allowing for adjustments to be made throughout the project lifecycle.	-It may be challenging to estimate accurately in an Agile environment, which can impact project timelines and budgets. -It may be difficult to prioritize features and requirements in a constantly changing environment.

				-It requires a high level of team involvement and commitment.
2022	F.Tahmasebinia et.al[31]	Monte Carlo Simulation	-Monte Carlo Simulation provides a probabilistic approach to project management, allowing for more accurate predictions and forecasts. -It can be used to analyze a wide range of variables and factors that impact project outcomes. -It can provide insight into the impact of different scenarios on project outcomes.	-Monte Carlo Simulation requires a significant amount of data to be effective, which may not always be available. -It can be time-consuming and complex to implement and interpret. The accuracy of the results can be impacted by the assumptions and models used.
2022	T.Wahyono et.al[33]	Use Case Points (UCP)	-Provides a simple and intuitive way to estimate the size and complexity of a project. -Allows for early stage estimation which enables early planning, budgeting and decision-making. -Can be used in combination with other estimation techniques.	-The accuracy of the estimation is dependent on the quality of the use cases identified. -The model might not be suitable for all types of projects. -The model might not take into account all the factors that influence project cost.
2022	N.A Avais Jan, Assad Abbas[36]	Earned Value Management (EVM)	-EVM provides a structured approach to project management, which can improve project performance. -It helps to identify project variances and deviations from the planned schedule and budget. -It can be used to forecast project completion dates and costs.	-EVM requires a significant amount of time and effort to implement and maintain. -It may not be suitable for smaller projects or projects with low budgets. -It can be difficult to understand and interpret the data produced by EVM.
2022	R. Wang et.al[39]	Neural Networks	-Capable of learning complex patterns and relationships from large datasets. -Can make predictions or classifications based on input data with high accuracy. -Can be used in a wide range of applications such	-Requires a large amount of data to train and validate the network. -The process of designing and training a neural network can be time-consuming and computationally intensive. -The internal workings of the network may be

			as image recognition, natural language processing, and predictive maintenance.	difficult to interpret or explain
2022	M.A.Salam[46]	Resource-Based Estimating	Allows for more accurate and detailed estimates, as it accounts for individual resources required for each task.	Can be time-consuming and difficult to implement for large projects. Can be affected by external factors that affect resource availability and cost.
2022	A. Kaur and K. Kaur[47]	COSMIC Function Points (CFP)	Provides a standardized, objective way to measure software functionality. Can be used to estimate project cost and duration accurately. Can be used across multiple platforms and software development methods.	Can be difficult to implement due to the complexity of the measurement process. Requires a detailed understanding of software functionality. Can be affected by the quality of the requirements specifications.
2022	X. Chen et.al[52]	Principal Component Analysis (PCA)	Can be used to identify underlying factors that affect multiple variables. Can be used to reduce the number of variables needed for analysis.	Assumes a linear relationship between variables, which may not always be accurate. May not provide a clear understanding of the underlying factors affecting the data.
2022	A.G Gad[63]	Particle Swarm Optimization (PSO)	Can optimize non-linear and non-convex functions with multiple variables Can converge to global optima with a relatively small number of iterations	Can be sensitive to the choice of parameters, such as the number of particles and the inertia weight optimization problems Does not guarantee convergence to the global optimum
2022	A.F Guven et.al[65]	Ant Colony Optimization (ACO)	Can find optimal or near-optimal solutions for complex problems Can handle problems with a large search space Can adapt to changing environments Easy to implement and parallelize.	Can be slow for large problems Requires a large number of iterations to converge to optimal or near-optimal solutions Can be sensitive to parameter settings, such as the evaporation rate and pheromone update rules

2022	M.F Ahmed et.al[67]	Differential Evolution	Can efficiently optimize high-dimensional and non-linear functions Does not require gradient information Easy to implement and can handle noisy functions Has a low computational cost	Might converge slowly to the optimal solution Difficult to tune the parameters Needs a large population to obtain accurate results Can suffer from premature convergence if parameters are not chosen correctly
2022	S. Liu et.al[70] L. Horvath et.al[23]	Firefly Algorithm	Can handle multimodal optimization problems with a large number of variables Converges faster than some other optimization algorithms	Might converge to suboptimal solutions Requires careful parameter tuning Can get stuck in local optima Needs a large population size for accurate results.
2021	S.Stappert et.al[24]	Parametric Cost Estimation	Provides a more accurate estimate of costs -Allows for the use of historical data and industry standards to generate estimates -Can be used for both initial and ongoing cost estimates	-Requires detailed knowledge of the project and its associated components -Can be costly and time consuming to implement
2021	KaeJang and W.J Kim[25]	Delphi method	-Provides an unbiased estimate of costs -Can be used with limited knowledge of the project -Can be completed in a short period of time	Results can be affected by personal biases of participants -Accuracy of results can be difficult to determine
2021	M.S. Islam et.al[32]	Expert Judgement	-Provides a structured way to gather expert opinions on project cost and time. -Can be used for both small and large projects.	-The accuracy of the estimation is dependent on the quality of the expert opinions gathered. -The model might not take into account all the factors that influence project cost
2021	W. Van Atteveldt et.al[34]	Three-Point Estimation	-Provides a more accurate estimate than a single-point estimation. -Takes into account the uncertainty and risk associated with the project. -Can be used in combination with other estimation techniques.	-The model assumes that the distribution of the estimates follows a certain pattern, which might not be true. -The accuracy of the estimation is dependent on the quality of the input data.

2021	J. Zhou et.al[35]	Analogous Estimating	-Provides a quick and easy way to estimate project cost and time. -Can be used when little information is available on the project. -Can be used in combination with other estimation techniques.	-The accuracy of the estimation is dependent on the quality of the historical data used. -The model might not take into account all the factors that influence project cost.
2021	A.N. Khan et.al[37]	Decision Tree Analysis	-Decision Tree Analysis provides a structured approach to decision-making, allowing for more informed and effective choices. -It can be used to analyze complex data and identify patterns and trends.	-Decision Tree Analysis can be time-consuming and complex to implement and interpret. -It may require a significant amount of data to be effective, which may not always be available.
2021	M. Forgione and Dipa[38]	Cost-Benefit Analysis (CBA)	-Provides a framework to evaluate the costs and benefits of a project or decision. -Helps in identifying the most feasible and cost-effective option. -Aids in the decision-making process by providing a quantitative assessment of the potential outcomes.	-The process of assigning values to intangible benefits or costs may be subjective. -The accuracy of the analysis depends heavily on the quality and completeness of the data used. -The analysis may not consider the broader social, environmental, or ethical implications of a decision.
2021	N. Rankovic et.al[40]	Artificial Intelligence (AI)	-Enables automation and optimization of complex tasks that were previously difficult or impossible to achieve. .-Can be applied in a wide range of domains such as healthcare, finance, and manufacturing.	-The use of AI raises ethical and social concerns such as bias, privacy, and job displacement. -The complexity of AI systems makes them difficult to regulate and control.
2021	A.H Ibrahim and L.M Elshwadfy[41]	Quality Function Deployment (QFD)	-Helps in translating customer requirements into specific product or service features. -Provides a structured approach to design and development. -Can help in identifying potential quality issues and	-The process can be time-consuming and resource-intensive. -The quality of the output is heavily dependent on the quality of the input data. -The approach may not be effective for highly

			improvement opportunities early in the process.	complex or dynamic products or services.
2021	A. Singh and S. Kumar[42]	Fuzzy logic estimation	-Fuzzy Logic Estimation provides a more nuanced approach to project estimation, taking into account uncertainty and imprecision. -It can provide insight into the impact of different scenarios on project outcomes.	-Fuzzy Logic Estimation can be complex and difficult to understand and interpret. The accuracy of the results can be impacted by the assumptions and models used.
2021	H.O Schmidt[43]	Program Evaluation and Review Technique (PERT)	Provides a probabilistic estimate of project duration and cost. Useful for managing complex or uncertain projects. Can help in identifying critical path activities and potential risks.	The accuracy of the estimate is heavily dependent on the quality of the input data. The approach may be time-consuming and resource-intensive. The use of a probabilistic estimate may be difficult for stakeholders to understand or accept.
2021	F. Elghaish et.al[44]	Design Structure Matrix (DSM)	Provides a visual representation of the relationships between different components or activities in a system. Useful for managing complex or interdependent projects.	The approach may not be suitable for highly dynamic or rapidly changing systems. The accuracy of the DSM is heavily dependent on the quality of the input data.
2021	A. Karimi and T.J Gandomani[45]	Function-Based Estimating	Easy to understand and implement. Provides a quick estimate of project cost and duration. Can be used for any type of project, regardless of complexity.	Highly dependent on the accuracy of assumptions and estimations made during the planning phase. Limited accuracy due to the inability to account for project-specific complexities.
2021	M.O. Sanni-Anibire et.al[48]	Simple Linear Regression	Provides a simple way to identify and analyze the relationship between two variables. Can provide insights into how changes in one variable affect another. .	Assumes a linear relationship between variables, which may not always be accurate. Does not account for other factors that may affect the relationship between variables.

2021	T.A. Trunfio et.al[49]	Multiple Regression Analysis	Allows for the analysis of the relationship between multiple variables simultaneously. Can account for the influence of other factors that may affect the relationship between variables.	Requires a significant amount of data to be effective. Assumes a linear relationship between variables, which may not always be accurate. May be affected by outliers or anomalies in the data.
2021	B.Liu,et.al[50]	Stepwise Regression	Can be used to identify the most important variables in a complex dataset. Can provide insights into how changes in one variable affect another. Can be used to make predictions about future outcomes based on historical data	May be affected by outliers or anomalies in the data. May miss important relationships between variables that are not included in the analysis. Can be time-consuming and computationally intensive.
2021	D. Lakens and A.R Caldwell[51]	Factor Analysis	Can be used to identify underlying factors that affect multiple variables. Can be used to reduce the number of variables needed for analysis.	Requires a significant amount of data to be effective.Assumes a linear relationship between variables, which may not always be accurate.
2021	Khan et.al[37]	Cluster Analysis	-Helps in identifying and grouping similar data points together, which can lead to insights and patterns that may not have been apparent before.	-Cluster analysis is highly dependent on the choice of clustering algorithm, and choosing the wrong algorithm can lead to inaccurate results.
2021	D.Gauman, S. Delgado and J.Perez[53]	Self-Organizing Maps (SOM)	-Can be used for a wide range of applications, including image recognition, data visualization, and feature extraction. -Can handle noisy and incomplete data without the need for data preprocessing.	-SOM requires a large amount of data to be effective, and small datasets may not produce accurate results. -SOM can be computationally expensive, especially when dealing with large datasets.
2021	D. Zhao et.al[54]	K-Nearest Neighbor (KNN)	-KNN is simple and easy to implement, making it a good choice for a wide range of applications.	-The choice of k can significantly affect the final results, KNN is sensitive to the choice of distance metric, and choosing the wrong metric can lead to inaccurate results.

			-KNN can handle both classification and regression problems.	
2021	A. Saeed et.al[55]	Top-Down Estimating	-Provides a quick and high-level estimate of project costs. -Useful for initial planning and budgeting. -Can help in identifying potential cost drivers and risks.	-The accuracy of the estimate may be low, especially for complex or uncertain projects. -The estimate may not be useful for detailed cost management or control.
2021	A. S. Filippetto et.al[56] V. Jackins et.al[57]	Design Structure Matrix (DSM)	Helps to visualize complex interdependencies between different components in a system or project Can help to manage complexity and improve efficiency in large-scale projects	Limited ability to capture temporal or causal relationships between components Can become unwieldy and difficult to interpret with large numbers of components
2021	M. Fan and A. Sharma[58]	Support Vector Machines (SVM)	Effective for classification and regression tasks, particularly with high-dimensional data Can be less prone to overfitting than some other models	Can be computationally expensive for large datasets Choice of kernel function and tuning parameters can have a significant impact on performance
2021	I.Wikramasinghe et.al[59]	Naive Bayes	Simple and computationally efficient Can work well with small datasets and high-dimensional feature spaces Can handle both categorical and continuous data	Assumes independence between features, which may not hold true in practice. Can be sensitive to irrelevant or noisy features Limited ability to capture complex relationships in the data
2021	P D'Souza and H D'Souza[60]	Bayesian Networks	Can capture complex relationships and dependencies between variables Can be used for both prediction and causal inference Provides a transparent and interpretable representation of the underlying model	Can be computationally expensive to build and evaluate May suffer from the "explaining away" problem, where evidence for one hypothesis reduces the probability of another
2021	B. Mor, S. Garhwal and A.Kumar[61]	Markov Models	Can model time-series data with a simple and transparent framework	Assumes the Markov property, where the current state depends only on the

			Can capture dependencies between observations at different time points Can be used for prediction and decision-making	previous state, which may not hold true in practice Can be sensitive to the choice of model order and assumptions about the underlying distribution
2021	C. Kim,R. Batra et.al[64]	Genetic Algorithms (GA)	Can handle complex and non-linear optimization problems Finds optimal or near-optimal solutions in a reasonable amount of time Can be applied to problems where little information is available about the problem domain	Requires a large number of evaluations to find optimal or near-optimal solutions Can be computationally expensive for large problem spaces Difficult to choose appropriate parameters, such as population size and mutation rate
2021	M. Fan and A. Sharma[58]	Simulated Annealing	Can escape local optima and explore a broader search space Can be used with any objective function or problem domain Works well with large, complex search spaces Can find the global optimum with high probability if given enough time	Can be computationally expensive, particularly for large search spaces Requires careful tuning of parameters Can get stuck in local optima if the cooling schedule is not chosen correctly
2021	C. Kahraman and E. Bolturk[66]	Hill Climbing	Simple and easy to implement Can be fast for small search spaces Finds the first local optimum quickly	Easily gets stuck in local optima Cannot handle multi-modal problems or non-convex optimization problems
2021	D.L. Loung et.al[68] M. Dubey et.al[69]	Evolutionary Programming (EP)	Works well for optimizing noisy functions Can handle problems with a large number of variables Can find global optima	Requires parameter tuning Can be slow to converge for complex problems Not efficient for problems with constraints.

4. Summary of the Comprehensive Approach for Software Cost Estimation.

a. Software Cost Estimation: The process of predicting the amount of effort, time, and resources required to develop or maintain software projects.

b. Intelligent Systems: Advanced computational systems that utilize machine learning, artificial intelligence, and data analytics to simulate human intelligence processes.

c. Machine Learning (ML): A subset of artificial intelligence that enables systems to learn and improve from experience without being explicitly programmed.

d. Artificial Intelligence (AI): The simulation of human intelligence processes by machines, especially computer systems, enabling them to perform tasks that typically require human intelligence.

e. COCOMO (Constructive Cost Model): A widely used software cost estimation model that calculates project costs based on historical data and project parameters.

f. Function Point Analysis (FPA): A method for measuring the size of software development based on the functionality delivered to the user, facilitating cost and effort estimation.

g. Predictive Analytics: The use of data, statistical algorithms, and machine learning techniques to identify the likelihood of future outcomes based on historical data.
h. Data-Driven Decision Making: The process of making decisions based on the analysis of data rather than intuition or observation alone.
i. Project Planning: The phase in the software development life cycle that involves defining project goals, scope, budget, timeline, and resources.
j. Resource Allocation: The process of assigning and managing assets in a manner that supports an organization's strategic goals.
k. Budgeting: The process of creating a plan to spend an organization's resources, outlining the estimated costs and revenues associated with a project.
l. Complexity: In the context of software development, refers to the degree of difficulty in understanding, designing, and managing a software project due to its size, structure, and interdependencies.
m. Adaptive Systems: Systems that are capable of adjusting their operation or structure in response to changes in the environment or feedback from the performance.
n. Efficiency: In software cost estimation, refers to the ability to achieve accurate estimations with minimal time, effort, and resources.
o. 15. Accuracy: The degree to which the result of a measurement, calculation, or specification conforms to the correct value or a standard.

5. Conclusion

The exploration of intelligent software cost estimation methods reveals a significant shift towards incorporating advanced analytics, machine learning, and artificial intelligence to tackle the inherent challenges of predicting software project costs accurately. Traditional models, while foundational, often fall short in addressing the dynamic and complex nature of contemporary software development projects. The comparative analysis of various estimation methods underscores the imperative for a more adaptive and data-driven approach. As the software industry continues to evolve, the integration of intelligent systems into cost estimation processes emerges as a critical factor for enhancing accuracy, efficiency, and project success. Future advancements in technology and methodology are expected to further refine and optimize cost estimation practices. This comprehensive overview highlights the necessity of embracing intelligent estimation methods, offering valuable insights for researchers, practitioners, and project managers aiming to navigate the complexities of software project planning and execution. By fostering a deeper understanding of these methods, the paper contributes to the ongoing efforts to improve project outcomes through more effective and informed cost estimation.

References

[1] E. Matel, F. Vahdatikhaki, S. Hosseinyalamdary, T. Evers, and H. Voordijk, “An artificial neural network approach for cost estimation of engineering services,” *Int. J. Constr. Manag.*, vol. 22, no. 7, pp. 1274–1287, 2022, doi: 10.1080/15623599.2019.1692400.

[2] S. Durdyev, “Review of construction journals on causes of project cost overruns,” *Eng. Constr. Archit. Manag.*, vol. 28, no. 4, pp. 1241–1260, 2020, doi: 10.1108/ECAM-02-2020-0137.

[3] A. Jadhav, M. Kaur, and F. Akter, “Evolution of Software Development Effort and Cost Estimation Techniques: Five Decades Study Using Automated Text Mining Approach,” *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/5782587.

[4] C. Proudfoot, R. Gautam, J. Cristino, R. Agrawal, L. Thakur, and K. Tolley, “Model parameters influencing the cost-effectiveness of sacubitril/valsartan in heart failure: evidence from a systematic literature review,” *Eur. J. Heal. Econ.*, no. 0123456789, 2022, doi: 10.1007/s10198-022-01485-3.

[5] S. A. Butt *et al.*, “A software-based cost estimation technique in scrum using a developer’s expertise,” *Adv. Eng. Softw.*, vol. 171, no. May, 2022, doi: 10.1016/j.advengsoft.2022.103159.

[6] C. Symons and M.C. Paulk, “Software Cost Estimation with Cocomo II,” *Int. Conf. Emerg. Trends Electr. Electron. Sustain. Energy Syst. ICETEESES 2016*, 2022.

[7] C. A. Ul Hassan *et al.*, “Optimizing Deep Learning Model for Software Cost Estimation Using Hybrid Meta-Heuristic Algorithmic Approach,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/3145956.

[8] S. Rajper and Z. A. Shaikh, “Software development cost estimation: A survey,” *Indian J. Sci. Technol.*, vol. 9, no. 31, pp. 177–205, 2016, doi: 10.17485/ijst/2016/v9i31/93058.

[9] s.k.shrivastava, “Software Cost Estimation: Principles and Practices,” 2021.

[10] L. X. b Yan Ji a, Zhen Kang a, Xiao Zhang b, “Model recovery for multi-input signal-output nonlinear systems based on the compressed sensing recovery theory,” *J. Franklin Inst.*, vol. Volume 359, no. Issue 5, March 2022, pp. 2317–2339, 2022.

[11] A. A. Fadhil, R. G. H. Alsarraj, and A. M. Altaie, “Software Cost Estimation Based on Dolphin Algorithm,” *IEEE Access*, vol. 8, pp. 75279–75287, 2020, doi: 10.1109/ACCESS.2020.2988867.

[12] S. M. R. Chirra and H. Reza, “A Survey on Software Cost Estimation Techniques,” *J. Softw. Eng. Appl.*, vol.

12, no. 06, pp. 226–248, 2019, doi: 10.4236/jsea.2019.126014.

[13] G. R. Reddy and H. Reza, "A Survey on Software Cost Estimation Techniques," *J. Softw. Eng. Appl.*, vol. 12, no. 06, pp. 226–248, 2022, doi: 10.4236/jsea.2019.126014.

[14] K. Bilal, K. Wahab, A. Muhammad, and J. Nazir, "A Survey on Software Cost Estimation Techniques," *Int. J. Data Sci. Adv. Anal.*, vol. 2, no. 2, pp. 1–5, 2020.

[15] A. Sidani *et al.*, "Recent tools and techniques of BIM-Based Augmented Reality: A systematic review," *J. Build. Eng.*, vol. 42, no. 0123456789, 2021, doi: 10.1016/j.jobe.2021.102500.

[16] M. Grinberg and J. Maldacena, "Proper time to the black hole singularity from thermal one-point functions," *J. High Energy Phys.*, vol. 2021, no. 3, 2021, doi: 10.1007/JHEP03(2021)131.

[17] J. Harnois-Déraps *et al.*, "Cosmic shear cosmology beyond two-point statistics: A combined peak count and correlation function analysis of DES-Y1," *Mon. Not. R. Astron. Soc.*, vol. 506, no. 2, pp. 1623–1650, 2021, doi: 10.1093/mnras/stab1623.

[18] M. Al-Azzawi, A.A.H., & Zubi, "Function Point Analysis for Software Quality Measurement," *Int. J. Comput. Appl. Technol. Res.*, vol. 183(4), 2021.

[19] M. Ben-Daya, "A comprehensive survey of Function Point Analysis," vol. 190(5), 2023.

[20] A. Akhtar, B. Bakhtawar, and S. Akhtar, "Extreme Programming Vs Scrum: a Comparison of Agile Models," *Int. J. Technol. Innov. Manag.*, vol. 2, no. 2, pp. 80–96, 2022, doi: 10.54489/ijtim.v2i2.77.

[21] B. Alsaadi and K. Saeedi, *Data-driven effort estimation techniques of agile user stories: a systematic literature review*, vol. 55, no. 7. Springer Netherlands, 2022. doi: 10.1007/s10462-021-10132-x.

[22] S. Obaid Barraood, H. Mohd, and F. Baharom, "A Comparison Study of Software Testing Activities in Agile Methods," *Semant. Sch.*, no. February, 2021, [Online]. Available: http://www.kmice.cms.net.my/

[23] L. Horváth, G. Rice, and Y. Zhao, "Change point analysis of covariance functions: A weighted cumulative sum approach," *J. Multivar. Anal.*, vol. 189, no. 2016, 2022, doi: 10.1016/j.jmva.2021.104877.

[24] S. Stappert, J. Wilken, G. J. D. Calabuig, and M. Sippel, "Evaluation of Parametric Cost Estimation in the Preliminary Design Phase of Evaluation of Parametric Cost Estimation in the Preliminary Design Phase of Reusable Launch Vehicles," no. July, 2022.

[25] K. ae Jang and W. J. Kim, "A method of activity-based software maintenance cost estimation for package software," *J. Supercomput.*, vol. 77, no. 8, pp. 8151–8171, 2021, doi: 10.1007/s11227-020-03610-6.

[26] R. Silhavy, P. Silhavy, and Z. Prokopova, "Using actors and use cases for software size estimation," *Electron.*, vol. 10, no. 5, pp. 1–21, 2021, doi: 10.3390/electronics10050592.

[27] E. M. D. B. Fávero, R. Pereira, A. R. Pimentel, and D. Casanova, "Analogy-based Effort Estimation: A Systematic Mapping of Literature," *Infocomp*, vol. 17, no. 2, pp. 7–22, 2018.

[28] K. D. D. Willis *et al.*, "JoinABLe: Learning Bottom-up Assembly of Parametric CAD Joints," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2022-June, pp. 15828–15839, 2022, doi: 10.1109/CVPR52688.2022.01539.

[29] R. Khoshfetrat, H. Sarvari, D. W. M. Chan, and M. Rakhshanifar, "Critical risk factors for implementing building information modelling (BIM): a Delphi-based survey," *Int. J. Constr. Manag.*, vol. 22, no. 12, pp. 2375–2384, 2022, doi: 10.1080/15623599.2020.1788759.

[30] U. K. Nath, A. K. Jagadev, P. K. Pattnaik, and S. K. Swain, "Effective Cost Estimation Using Agile Process," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 18, pp. 5209–5219, 2022.

[31] F. Tahmasebinia, R. Jiang, S. Sepasgozar, J. Wei, Y. Ding, and H. Ma, "Implementation of BIM Energy Analysis and Monte Carlo Simulation for Estimating Building Energy Performance Based on Regression Approach: A Case Study," *Buildings*, vol. 12, no. 4, 2022, doi: 10.3390/buildings12040449.

[32] M. S. Islam, M. P. Nepal, M. Skitmore, and R. Drogemuller, "Risk induced contingency cost modeling for power plant projects," *Autom. Constr.*, vol. 123, 2021, doi: 10.1016/j.autcon.2020.103519.

[33] T. Wahyono, Purwanto, E. Maria, and K. D. Hartomo, "Software Complexity Analysis of Integrated Accounting Information Systems for Cooperatives using the Use Case Point (UCP) Method," in *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)*, 2022, pp. 1–6. doi: 10.1109/ICOSNIKOM56551.2022.10034924.

[34] W. van Atteveldt, M. A. C. G. van der Velden, and M. Boukes, "The Validity of Sentiment Analysis: Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms," *Commun. Methods Meas.*, vol. 15, no. 2, pp. 121–140, 2021, doi: 10.1080/19312458.2020.1869198.

[35] J. Zhou *et al.*, "Predicting TBM penetration rate in hard rock condition: A comparative study among six XGB-based metaheuristic techniques," *Geosci. Front.*, vol. 12, no. 3, p. 101091, 2021, doi: 10.1016/j.gsf.2020.09.020.

[36] N. A. Avais Jan, Assad Abbas, "Monitoring and Controlling Software Project Scope Using Agile EVM," *Wiley Online Libr.*, no. https://doi.org/10.1002/9781119821779.ch4, 2022.

[37] A. N. Khan, N. Iqbal, A. Rizwan, R. Ahmad, and D. H. Kim, “An ensemble energy consumption forecasting model based on spatial-temporal clustering analysis in residential buildings,” *Energies*, vol. 14, no. 11, pp. 1–25, 2021, doi: 10.3390/en14113020.

[38] M. Forgione and D. Piga, “Continuous-time system identification with neural networks: Model structures and fitting criteria,” *Eur. J. Control*, vol. 59, pp. 69–81, 2021, doi: 10.1016/j.ejcon.2021.01.008.

[39] R. Wang, V. Asghari, C. M. Cheung, S. C. Hsu, and C. J. Lee, “Assessing effects of economic factors on construction cost estimation using deep neural networks,” *Autom. Constr.*, vol. 134, no. February, 2022, doi: 10.1016/j.autcon.2021.104080.

[40] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, “A New Approach to Software Effort Estimation Using Different Artificial Neural Network Architectures and Taguchi Orthogonal Arrays,” *IEEE Access*, vol. 9, pp. 26926–26936, 2021, doi: 10.1109/ACCESS.2021.3057807.

[41] A. H. Ibrahim and L. M. Elshwadfy, “Factors affecting the accuracy of construction project cost estimation in Egypt,” *Jordan J. Civ. Eng.*, vol. 15, no. 3, pp. 329–344, 2021.

[42] A. Singh and S. Kumar, “Picture fuzzy set and quality function deployment approach based novel framework for multi-criteria group decision making method,” *Eng. Appl. Artif. Intell.*, vol. 104, no. July, p. 104395, 2021, doi: 10.1016/j.engappai.2021.104395.

[43] [43] H. O. Schmidt, *Construction Cost Estimating International.* Routledge, 1977.

[44] [44] F. Elghaish, S. Abrishami, M. R. Hosseini, and S. Abu-Samra, “Revolutionising cost structure for integrated project delivery: a BIM-based solution,” *Eng. Constr. Archit. Manag.*, vol. 28, no. 4, pp. 1214–1240, 2021, doi: 10.1108/ECAM-04-2019-0222.

[45] [45] A. Karimi and T. J. Gandomani, “Software development effort estimation modeling using a combination of fuzzy-neural network and differential evolution algorithm,” *Int. J. Electr. Comput. Eng.*, vol. 11, no. 1, pp. 707–715, 2021, doi: 10.11591/ijece.v11i1.pp707-715.

[46] M. A. Salam, “Analyzing manufacturing strategies and Industry 4.0 supplier performance relationships from a resource-based perspective,” *Benchmarking*, vol. 28, no. 5, pp. 1697–1716, 2022, doi: 10.1108/BIJ-12-2018-0428.

[47] A. Kaur and K. Kaur, “A COSMIC function points based test effort estimation model for mobile applications,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 3, pp. 946–963, 2022, doi: 10.1016/j.jksuci.2019.03.001.

[48] M. O. Sanni-Anibire, R. Mohamad Zin, and S. O. Olatunji, “Developing a preliminary cost estimation model for tall buildings based on machine learning,” Int. J. Manag. Sci. Eng. Manag., vol. 16, no. 2, pp. 134–142, 2021, doi:10.1080/17509653.2021.1905568.

[49] T. A. Trunfio, A. Scala, A. Della Vecchia, A. Marra, and A. Borrelli, “Multiple Regression Model to Predict Length of Hospital Stay for Patients Undergoing Femur Fracture Surgery at ‘San Giovanni di Dio e Ruggi d’Aragona’ University Hospital,” *IFMBE Proc.*, vol. 80, pp. 840–847, 2021, doi: 10.1007/978-3-030-64610-3_94.

[50] B. Liu, Q. Zhao, Y. Jin, J. Shen, and C. Li, “Application of combined model of stepwise regression analysis and artificial neural network in data calibration of miniature air quality detector,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–12, 2021, doi: 10.1038/s41598-021-82871-4.

[51] D. Lakens and A. R. Caldwell, “Simulation-Based Power Analysis for Factorial Analysis of Variance Designs,” *Adv. Methods Pract. Psychol. Sci.*, vol. 4, no. 1, 2021, doi: 10.1177/2515245920951503.

[52] X. Chen, J. D. Lee, H. Li, and Y. Yang, “Distributed Estimation for Principal Component Analysis: An Enlarged Eigenspace Analysis,” *J. Am. Stat. Assoc.*, vol. 117, no. 540, pp. 1775–1786, 2022, doi: 10.1080/01621459.2021.1886937.

[53] D. Guaman, S. Delgado, and J. Perez, “Classifying Model-View-Controller Software Applications Using Self-Organizing Maps,” *IEEE Access*, vol. 9, pp. 45201–45229, 2021, doi: 10.1109/ACCESS.2021.3066348.

[54] D. Zhao *et al.*, “k-means clustering and kNN classification based on negative databases,” *Appl. Soft Comput.*, vol. 110, p. 107732, 2021, doi: 10.1016/j.asoc.2021.107732.

[55] A. Saeed, W. H. Butt, F. Kazmi, and M. Arif, “Survey of software development effort estimation techniques,” *ACM Int. Conf. Proceeding Ser.*, pp. 82–86, 2021, doi: 10.1145/3185089.3185140.

[56] A. S. Filippetto, R. Lima, and J. L. V. Barbosa, “A risk prediction model for software project management based on similarity analysis of context histories,” *Inf. Softw. Technol.*, vol. 131, no. March 2020, 2021, doi: 10.1016/j.infsof.2020.106497.

[57] V. Jackins, S. Vimal, M. Kaliappan, and M. Y. Lee, “AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes,” *J. Supercomput.*, vol. 77, no. 5, pp. 5198–5219, 2021, doi: 10.1007/s11227-020-03481-x.

[58] M. Fan and A. Sharma, “Design and implementation of construction cost prediction model based on SVM and LSSVM in industries 4.0,” *Int. J. Intell. Comput. Cybern.*, vol. 14, no. 2, pp. 145–157, Jan. 2021, doi: 10.1108/IJICC-10-2020-0142.

[59] I. Wickramasinghe and H. Kalutarage, “Naive Bayes: applications, variations and vulnerabilities: a review of

literature with code snippets for implementation," *Soft Comput.*, vol. 25, no. 3, pp. 2277–2293, 2021, doi: 10.1007/s00500-020-05297-6.

[60] D. D'Souza and H. D'Souza, "Bilingual adaptations in early development," *Trends Cogn. Sci.*, vol. 25, no. 9, pp. 727–729, 2021, doi: 10.1016/j.tics.2021.06.002.

[61] B. Mor, S. Garhwal, and A. Kumar, "A Systematic Review of Hidden Markov Models and Their Applications," *Arch. Comput. Methods Eng.*, vol. 28, no. 3, pp. 1429–1448, 2021, doi: 10.1007/s11831-020-09422-4.

[62] R. Glennie, T. Adam, V. Leos-Barajas, T. Michelot, T. Photopoulou, and B. T. McClintock, "Hidden Markov models: Pitfalls and opportunities in ecology," *Methods Ecol. Evol.*, vol. 14, no. 1, pp. 43–56, 2023, doi: 10.1111/2041-210X.13801.

[63] A. G. Gad, *Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review*, vol. 29, no. 5. Springer Netherlands, 2022. doi: 10.1007/s11831-021-09694-4.

[64] C. Kim, R. Batra, L. Chen, H. Tran, and R. Ramprasad, "Polymer design using genetic algorithm and machine learning," *Comput. Mater. Sci.*, vol. 186, pp. 1–15, 2021, doi: 10.1016/j.commatsci.2020.110067.

[65] A. F. Güven, N. Yörükeren, and M. M. Samy, "Design optimization of a stand-alone green energy system of university campus based on Jaya-Harmony Search and Ant Colony Optimization algorithms approaches," *Energy*, vol. 253, p. 124089, 2022, doi: https://doi.org/10.1016/j.energy.2022.124089.

[66] C. Kahraman and E. Bolturk, *Toward Humanoid Robots: The Role of Fuzzy Sets*. 2021.

[67] M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution: A recent review based on state-of-the-art works," *Alexandria Eng. J.*, vol. 61, no. 5, pp. 3831–3872, 2022, doi: 10.1016/j.aej.2021.09.013.

[68] D. L. Luong, D. H. Tran, and P. T. Nguyen, "Optimizing multi-mode time-cost-quality trade-off of construction project using opposition multiple objective difference evolution," *Int. J. Constr. Manag.*, vol. 21, no. 3, pp. 271–283, 2021, doi: 10.1080/15623599.2018.1526630.

[69] M. Dubey, V. Kumar, M. Kaur, and T. P. Dao, "A Systematic Review on Harmony Search Algorithm: Theory, Literature, and Applications," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/5594267.

[70] S. Liu, Z. Chuang, K. Wang, X. Li, X. Chang, and L. Hou, "Structural Parametric Optimization of the VolturnUS-S Semi-Submersible Foundation for a 15 MW Floating Offshore Wind Turbine," *J. Mar. Sci. Eng.*, vol. 10, no. 9, 2022, doi: 10.3390/jmse10091181.

[71] J. Cheng, Y. Wang, J. H. Park, J. Cao, and K. Shi, "Static Output Feedback Quantized Control for Fuzzy Markovian Switching Singularly Perturbed Systems With Deception Attacks," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 4, pp. 1036–1047, 2022, doi: 10.1109/TFUZZ.2021.3052104.

[72] S. O. Abioye *et al.*, "Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges," *J. Build. Eng.*, vol. 44, no. April 2020, p. 103299, 2021, doi: 10.1016/j.jobe.2021.103299.

[73] [J. A. Khan, S. U. R. Khan, T. A. Khan, and I. U. R. Khan, "An Amplified COCOMO-II Based Cost Estimation Model in Global Software Development Context," *IEEE Access*, vol. 9, pp. 88602–88620, 2021, doi: 10.1109/ACCESS.2021.3089870.

[74] F. Afzal, S. Yunfei, M. Nazir, and S. M. Bhatti, "A review of artificial intelligence based risk assessment methods for capturing complexity-risk interdependencies: Cost overrun in construction projects," *Int. J. Manag. Proj. Bus.*, vol. 14, no. 2, pp. 300–328, 2021, doi: 10.1108/IJMPB-02-2019-0047.